In the Claims:

1. (cancelled) An utterance detector comprising:

a frame-level detector for making speech/non-speech decisions for each frame, and

an utterance detector coupled to said frame-level detector and responsive to said speech/non-speech decisions over a period of frames to detect an utterance.

2. (cancelled) The utterance detector of Claim 1, wherein said utterance level detector is a state machine.

3. (cancelled) The utterance detector of Claim 2, wherein said state machine has the states of pre-speech, non-speech, in-speech and pre-non-speech.

4. (currently amended): An utterance detector comprising:

a frame-level detector for making speech/non-speech decisions for each frame, and

an utterance detector coupled to said frame-level detector and responsive to said speech/non-speech decisions over a period of frames to detect an utterance;

~~The utterance detector of Claim 1, wherein~~ said frame-level detector includes autocorrelation.

5. (currently amended) An utterance detector comprising:

a frame-level detector for making speech/non-speech decisions for each frame, and

an utterance detector coupled to said frame-level detector and responsive to said speech/non-speech decisions over a period of frames to detect an utterance; said frame-

level detector includes autocorrelation;~~The~~ said utterance detector ~~of Claim 4,~~ including filter means for performing frequency-selective autocorrelation.

6. (original) The utterance detector of Claim 5, wherein said autocorrelation and filtering is performed in DFT domain by taking the signal and applying DFT, performing frequency domain windowing and then inverse DFT.

7. (original) The utterance detector of Claim 1, wherein said frame-level frame detector includes means for calculating power spectrum of an input signal, performing frequency shaping, performing inverse FFT and determining maximum value of periodicity.

8. (original) The utterance detector of Claim 7, wherein calculating power spectrum includes the steps of filtering the signal, applying a Hamming window and performing FFT on the signal from the Hamming window.

9. (currently amended) The utterance detector of Claim 7, wherein said performing frequency shaping step includes the step of:

$$F(k)=\begin{cases}\alpha^{F_l-k} & \text{if} \quad 0\le k<F_l \\ 1 & \text{if} \quad F_l\le k<F_h \\ \beta^{k-F_h} & \text{if} \quad F_h\le k<\dfrac{N}{2}\end{cases}$$

where $F_l$ and $F_h$ are low and high frequency indices respectfully. R(k) is the autocorrelation, $F(k)$ is a filter, and $\alpha$ and $\beta$ are constants

with $\alpha$=0.70

$\beta$=0.85

to get R(k).

In the Specification:

Please amend the specification as follows:

On page 2 the sixth paragraph ( lines 10 and 11 ) change to:

FIG. 7A is a time signal (non-speech portion ) and FIG. 7B illustrates frequency-selective autocorrelation function of the time signal of FIG. 7A;

On page 3 third full paragraph ( lines 13-19) change to:

For resistance to noise, Applicants teach to exploit the periodicity, rather than energy, of the speech signal. Specifically, we use autocorrelation function. The autocorrelation function (correlation with signal delayed by $\tau$ ) used in this work is derived from speech *X(t),* and is defined as:

$$R_x(\tau) = E[X(t)X(t+\tau)] \qquad (1)$$

Important properties of $Rx(\tau)$ include:

$$R_x(0) \geq R_x(\tau). \qquad (2)$$

$$R_x(\tau) = R_S(\tau) + R_N(\tau)$$

On page 4, second full paragraph ( lines 7 -14) change to:

The autocorrelation is for signal plus noise as represented in FIG. 4. Most random noise signals are not correlated, *i.e.*, they satisfy:

$$\lim_{\tau \to \infty} R_N(\tau) = 0. \qquad (5)$$

This is represented by autocorrelation in FIG 5 as zero. Therefore, we have for large $\tau$ :

$$R_x(\tau) \approx R_s(\tau) \qquad (6)$$

Therefore, for large $T$, the noise has no correlation function. This property says that autocorrelation function has some noise immunity.

On page 5, first full paragraph to end of the page (lines 3-21) change to:

We apply a filter $f(\tau)$ on the power spectrum of the autocorrelation function to attenuate the above-mentioned undesirable noisy components, as described by:

$$r_x(\tau) = R_x(\tau) * f(\tau) \qquad (7)$$

To reduce the computation as in equation 1 and equation 7, the convolution is performed in the Discrete Fourier Transform (DFT) domain, as detailed below in the implementation. We can do the same by a DFT as illustrated in FIG. 6 by taking the signal and applying DFT, then do a frequency domain windowing following the equation 8 below and then do an inverse DFT to get the autocorrelation. The filter $f(\tau)$ is specified in the frequency domain:

$$F(k) = \begin{cases} \alpha^{F_l - k} & \text{if} \quad 0 \le k < F_l \\ 1 & \text{if} \quad F_l \le k < F_h \\ \beta^{k - F_h} & \text{if} \quad F_h \le k < \dfrac{N}{2} \end{cases} \qquad (8)$$

with $\alpha = 0.70$ (9)

$\beta = 0.85$ (10)

where $F_l$ and $F_h$ are respectively the discrete frequency indices under given sample frequency for 600 Hz and 1800 Hz.

We show two plots of $r_X(\tau)$ along with the time signal. The signal has been corrupted to 0 dB SNR. FIG. 7A shows a non-speech signal and FIG. 7B the frequency selective autocorrelation of the non-speech signal. FIG. 8A shows a speech signal and FIG. 8B the frequency selective autocorrelation function. It can be seen for the speech signal, a peak at 60 in FIG. 8B can be detected, with an amplitude substantially stronger than any peak in FIG. 7B.

On pages 7 and 8 beginning on the last paragraph of page 7 and continuing on page 8 ( page 7, line 23 through page 8, line12) change to:

In FIG. 12, one cycle means state. The arrow means to go to another state. The numbers are paths. Each path is defined by a condition. These are from level decisions. For each path, we need to take an action. Actions include state transitions. The action can be to do some calculation. After that action, we make a transition to the next state. In Table 1, the state is indicated by case. Suppose we need to make an utterance decision. We have four cases on states which are non-speech, pre-speech, in-speech and pre-nonspeech. We initialize on the left most case which is non-speech. We look at input. If the input frame is speech, we initialize a counter ($n$ = 1). In this case, we go to pre-speech state via path 2. If the frame level is non-speech, the system stays in the same state as represented by path 1. If in the pre-speech state and there is not enough counts of frames to indicate in-speech yet the frame is indicated or speech, we stay in pre-speech and increase the count by 1 as indicated by path 4. If the frame is speech and the count is $N$ or greater (sufficiently long time), then it goes to the in-speech state as indicated by path 5. If the frame is not speech, then it takes the path 3 back to non-speech state. If we continue to detect speech at the frame level, we stay in the same state (~~patch~~ path 6). If we receive a non-speech frame we move to pre-nonspeech state (path 7). If we again observe speech, we go back to in-speech state (path 8). If the next frame is non-speech, we stay in pre-nonspeech (path 9). If in pre-nonspeech for sufficiently long time (count of $N$) and frame input is below threshold, then we are in non-speech and the system goes to the non-speech state (path 10).